

**Analisa Bahasa Pemrograman Gentee Dengan Bahasa C++
Program Aplikasi Perkalian Matriks**



Skripsi

Diajukan untuk Memenuhi salah Satu Syarat dalam Meraih Gelar Sarjana Sains
Jurusan Matematika pada Fakultas Sains dan Teknologi
Universitas Islam Negeri Alauddin Makassar

Oleh

ULFA EKA SAPUTRI

NIM. 60600108017

UNIVERSITAS ISLAM NEGERI

ALAUDDIN

M A K A S S A R

JURUSAN MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI ALAUDDIN MAKASSAR

2012

DAFTAR ISI

	halaman
HALAMAN SAMPUL	i
PERNYATAAN KEASLIAN SKRIPSI.....	ii
PERSETUJUAN PEMBIMBING	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
ABSTRAK	x
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	4
C. Tujuan Penelitian.....	4
D. Batasan Masalah.....	4
E. Manfaat Penelitian.....	4
F. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	6
A. Algoritma.....	6
B. Flowchart.....	7
C. Bahasa Pemrograman C++.....	9
D. Dasar Pemrograman Gentee.....	13
E. Kriteria Perbandingan Bahasa Program.....	24
F. Matriks.....	25
BAB III METODE PENELITIAN.....	29
A. Jenis Penelitian.....	29

B. Sumber Data.....	29
C. Lokasi Penelitian	29
D. Prosedur Penelitian	29
 BAB IV HASIL DAN PEMBAHASAN	32
A. Hasil.....	32
B. Pembahasan.....	42
 BAB V PENUTUP	47
A. Kesimpulan.....	47
B. Saran	49
 DAFTAR PUSTAKA.....	50
LAMPIRAN.....	51

DAFTAR GAMBAR

Gambar 1. Algoritma Memasak Air.....	8
Gambar 2. Struktur Program C++.....	9
Gambar 3. Pendeklarasian Variabel.....	10
Gambar 4. Program <i>Hello World</i>	13
Gambar 5. Matriks.....	19
Gambar 6. Flowchart Perkalian Matriks.....	25
Gambar 7. Aplikasi Perkalian Matriks C++.....	27
Gambar 8. Aplikasi Perkalian Matriks Gentee.....	30
Gambar 10. Output Aplikasi Perkalian Matriks C++.....	35
Gambar 11. Output Aplikasi Perkalian Matriks Gentee.....	36
Gambar 12. <i>Output Test Normality</i>	38
Gambar 13. Output Independent Sample Test`.....	39

DAFTAR TABEL

Tabel 1. Simbol Flowchart.....	7
Tabel 2. Hasil Eksekusi Program Aplikasi Perkalian Matriks.....	37



Abstrack

Name : Ulfa Eka Saputri
Nim : 60600108017
Title : Analysis Gentee Programing language and C++ Programing language Matrix Multiplication Program Application.

The more rapid development of technology at this time spawned a variety of languages programming, with many of the birth of the programming language certainly has its pros and cons of each. Through this scientific writing, will be analyzed comparing the performance of the Gentee programming language with C++. Examples of matrix multiplication application program to be used as a medium of both languages so the analysis will get some facts about the advantages and disadvantages of these two programming languages. Criteria for comparison of the performance of a programming language by analyzing syntax program of both languages, program execution speed using Apptimer software, and memory usage.

Research found that the memory usage on Gentee matrix multiplication program is 3 Kb and average execution speed is milolisecond 0.0235, while the memory usage in C++ matrix multiplication program is 540 Mb and the average execution rate is 0.0338 milolisecond.

Keyword : Gentee Programing language, C++ Programing language

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

Abstrak

Nama : Ulfa Eka Saputri
Nim : 60600108017
**Judul Skripsi : Analisa Bahasa Pemrograman Gentee dan Bahasa C++
Program Aplikasi Perkalian Matriks.**

Makin pesatnya perkembangan teknologi saat ini melahirkan berbagai bahasa pemrograman, dengan banyak lahirnya bahasa pemrograman tersebut tentunya memiliki kelebihan dan kekurangannya masing-masing. Melalui penulisan ilmiah ini, akan dianalisa perbandingan kinerja dari bahasa pemrograman Gentee dengan C++. Contoh program Aplikasi Perkalian Matriks yang akan digunakan sebagai media analisa kedua bahasa tersebut sehingga akan didapat suatu fakta mengenai kelebihan dan kekurangan kedua bahasa pemrograman tersebut. Kriteria perbandingan kinerja bahasa pemrograman dengan menganalisa syntax program dari kedua bahasa tersebut, kecepatan eksekusi program menggunakan software Apptimer, dan penggunaan memori.

Hasil Penelitian diperoleh bahwa penggunaan memori pada Gentee Aplikasi Perkalian Matriks adalah 3 Kb dan kecepatan eksekusi rata-rata adalah 0.0235 milolisecond, sedangkan penggunaan memori pada C++ Aplikasi Perkalian Matriks adalah 540 Kb dan kecepatan eksekusi rata-rata adalah 0.0338 milolisecond.

Kata Kunci : Pemrograman Gentee, Pemrograman C++

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi Informasi dan Komunikasi (TIK) telah berkembang sangat jauh saat ini dan telah merevolusi cara hidup, baik terhadap cara berkomunikasi, cara belajar, cara bekerja, cara berbisnis, dan lain sebagainya. Era informasi memberikan ruang lingkup yang sangat besar untuk mengorganisasikan segala kegiatan melalui cara baru, inovatif, instan, transparan, akurat, tepat waktu, lebih baik, memberikan kenyamanan yang lebih dalam mengelola dan menikmati kehidupan. Dengan teknologi informasi dan komunikasi semua proses kerja dan konten akan ditransformasikan dari fisik dan statis menjadi digital, mobile, virtual dan personal. Akibatnya kecepatan kinerja bisnis meningkat dengan cepat. Kecepatan proses meningkat sangat tajam di banyak aktivitas modern manusia. Teknologi Informasi adalah sarana/prasarana, sistem dan metode untuk perolehan, pengiriman, penerimaan, pengolahan, dan penapsiran, penyimpanan, pengorganisasian, dan penggunaan data yang bermakna. Teknologi informasi juga dapat dikatakan suatu teknologi yang digunakan untuk mengolah data, termasuk memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas, yaitu informasi yang relevan, akurat dan tepat waktu,

yang digunakan untuk keperluan pribadi, pendidikan, bisnis, dan pemerintahan dan merupakan informasi yang strategis untuk pengambilan keputusan.¹

Dalam ajaran islam dijelaskan bahwa nabi adalah sumber informasi dari Tuhan kepada manusia, bukan sekedar menerima wahyu dan kemudian menyampaikannya, tetapi sekaligus juga menjelaskan kehendak tuha n dengan bahasa manusia sehingga bias dimengerti oleh umatnya. Allah berfirman dalam Al-Qur'an Surah Asy Syuraa ayat 51.

﴿ وَمَا كَانَ لِنَبِيٍّ أَنْ يَكَلِّمَهُ اللَّهُ إِلَّا وَحْيًا أَوْ مِنْ وَرَآيِ حِجَابٍ أَوْ يُرْسِلَ رَسُولًا فَيُوحِيَ بِإِذْنِهِ مَا يَشَاءُ إِنَّهُ عَلَىٰ حَكِيمٍ مُّبِينٌ ﴾

Terjemahan :

Dan tidak mungkin bagi seorang manusiapun bahwa Allah berkata-kata dengan dia kecuali dengan perantaraan wahyu atau dibelakang tabir atau dengan mengutus seorang utusan (malaikat) lalu diwahyukan kepadanya dengan seizin-Nya apa yang dia kehendaki. Sesungguhnya dia Maha Tinggi lagi Maha Bijaksana.²

Saat ini teknologi informasi sangat dibutuhkan oleh berbagai aktifitas manusia sehari-hari. Sebagai contoh, dalam bidang perbankan yang mungkin tak akan lepas dari teknologi informasi. Berbagai sistem maupun aplikasi yang digunakan pada bidang ini, dikembangkan dari berbagai macam bahasa pemrograman.

¹ Fadlibae Weblog's. "Latar Belakang Pengembangan Teknologi Informasi Dan Komunikasi Untuk Pendidikan". <http://fadlibae.wordpress.com/2010/08/24/latar-belakang-pengembanganteknologi-informasi-dan-komunikasi-untuk-pendidikan>. (2 Januari 2012)

² Departemen Agama RI, *Al-Quran dan terjemahannya* (Semarang : PT Syifa, 1988), h.286

Bahasa pemrograman yang digunakan untuk mengembangkan sistem maupun aplikasi terus berkembang sejak dikembangkan bahasa pemrograman untuk pertama kali. Dimulai dari bahasa pemrograman BASIC yang bersifat modular, kemudian muncul bahasa PASCAL, FORTRAN, C, dan COBOL sebagai bahasa pemrograman yang bersifat terstruktur. Sekarang ini telah muncul lagi metodologi pemrograman yang banyak sekali diimplementasikan keberbagai aplikasi atau sistem, yaitu metodologi pemrograman berorientasi objek dengan bahasa pemrogramannya yaitu C++, Java, PHP, dan lain-lain.

Berbagai bahasa pemrograman yang disebutkan di atas adalah bahasa pemrograman yang banyak digunakan oleh para pengembang untuk mengembangkan sistem maupun aplikasi. Selain dari bahasa pemrograman Pascal, C, C++, Java dan PHP. Ada juga bahasa pemrograman baru yang mungkin agak asing yaitu D, Nice, Boo, Dao dan Gentee.

Sekian banyak bahasa pemrograman baru tersebut termasuk di dalamnya adalah bahasa Gentee. Bahasa Gentee adalah bahasa pemrograman yang terstruktur dan *open source*, sehingga dalam waktu dekat ini bahasa Gentee mungkin akan terus berkembang³. Sintaks-sintaks dasar serta struktur pemrograman pada bahasa Gentee hampir mirip dengan bahasa C++.

Alasan penulis memilih bahasa pemrograman Gentee, selain sifatnya yang “open source”, ukuran software yang tidak terlalu besar (3,1 MB) serta memiliki struktur bahasa pemrograman yang layak untuk dipelajari. .

³ Gentee. “Gentee Overview Introduction”. <http://www.gentee.com>.(2 Januari 2012).

B. Rumusan Masalah

Berdasarkan latar belakang tersebut maka perumusan masalah yang diambil dalam penelitian ini adalah :

1. Bagaimana membuat aplikasi perhitungan perkalian matriks dengan bahasa pemrograman Gentee dan bahasa pemrograman C++ ?
2. Bagaimana perbandingan bahasa pemrograman Gentee dan bahasa pemrograman C++ dengan pembuatan aplikasi yang sama ?

C. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Membuat aplikasi perkalian matriks dengan bahasa pemrograman Gentee dan bahasa pemrograman C++.
2. Membandingkan bahasa pemrograman Gentee dan bahasa pemrograman C++ dengan pembuatan aplikasi yang sama.

D. Batasan Masalah

Dalam hal ini permasalahan yang dibahas terbatas pada perkalian matriks dengan menggunakan bahasa pemrograman Gentee dan sekaligus diperbandingkan dengan menggunakan bahasa pemrograman C++.

E. Manfaat Penelitian

Adapun manfaat yang bisa diperoleh dari penelitian ini antara lain sebagai berikut:

1. Bagi peneliti

Peneliti menjadikan hasil penelitian ini sarana evaluasi terhadap kemampuan dalam mengaplikasikan teori-teori pemrograman bahasa C++ dan bahasa Gentee.

2. Bagi jurusan

Dengan adanya penelitian ini, jurusan dapat menjadikannya sebagai bahan referensi teori-teori pemrograman bahasa C++ dan bahasa Gentee.

3. Bagi Pembaca

Dengan adanya penelitian ini, pembaca dapat menambah pengetahuan dan wawasan mengenai pemrograman bahasa C++ dan bahasa Gentee.

F. Sistematika Penulisan

Secara umum, penelitian ini terdiri dari lima bab: bab I pendahuluan yang berisikan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah dan manfaat penelitian serta sistematika penulisan. Bab II tinjauan pustaka, berisikan konsep-konsep mengenai, algoritma, flowchart, bahasa pemrograman C++, bahasa pemrograman Gentee dan matriks. Bab III metodologi penelitian, berisikan jenis penelitian, sumber data, lokasi penelitian dan prosedur penelitian. Bab IV hasil dan pembahasan. Bab V penutup, berisikan kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Algoritma

Algoritma merupakan pola pikir yang terstruktur yang berisi tahap-tahap penyelesaian suatu masalah, yang nantinya akan diimplementasikan ke dalam suatu bahasa pemrograman.⁴

Algoritma juga merupakan jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam *terminology* algoritma. Dalam kehidupan sehari-hari banyak terdapat proses yang digambarkan dalam suatu algoritma. Cara-cara membuat kue atau masakan, misalnya dinyatakan dalam suatu resep. Resep adalah suatu algoritma.⁵

Berikut adalah contoh dari algoritma memasak air :

Gambar 1. Algoritma Memasak Air

1. Mulai
2. Menyiapkan peralatan masak air
3. Menyalakan kompor minyak tanah.
4. Memasak air di kompor yang sudah menyala.
5. Menunggu hingga air masak.
6. Selesai.

⁴ Andri Kristanto, *Algoritma dan Pemrograman dengan C++* (Cet, 2; Yogyakarta: Graha Ilmu, 2009), h. 9.

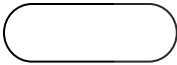



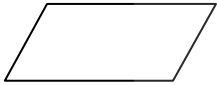

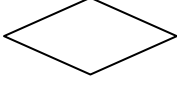
⁵ Rinaldi Munir, *Algoritma Dan Pemrograman Dalam Bahasa Pascal dan C* (Cet, 6; Bandung : Informatika Bandung, 2005), h.11.

B. Flowchart



Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.⁶

Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart :

Tabel 1. Simbol Flowchart

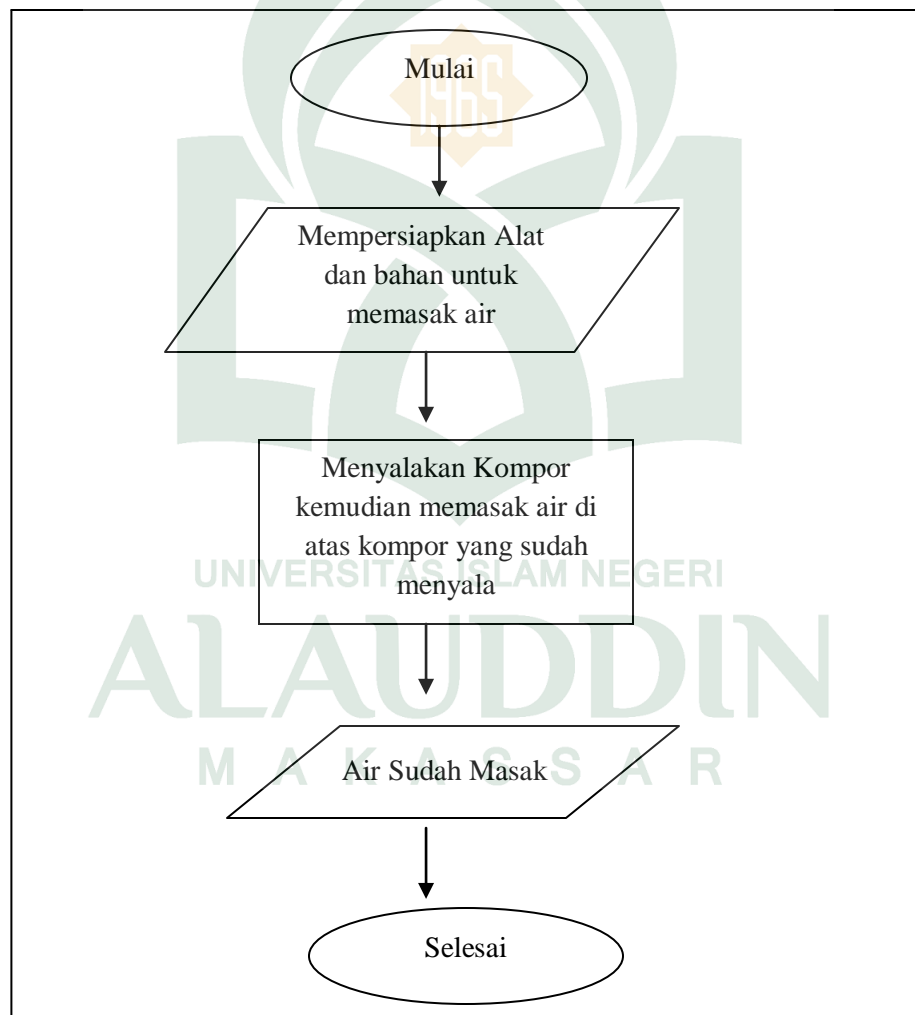
SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

⁶ Petra, "Algoritma dan Flowchart," <http://faculty.petra.ac.id/thiang/download/dkp/Algoritma%20dan%20Flowchart.doc> (02 Januari 2012).

	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Berikut adalah contoh dari algoritma memasak air yang di bahas pada materi algoritma :

Gambar 1. Flowchart Memasak air



C. Bahasa Pemrograman C++

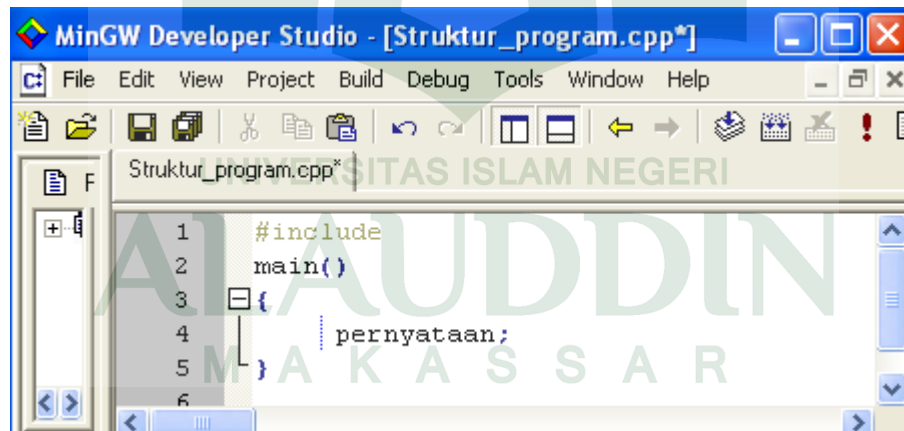
Bahasa C dan C++ merupakan bahasa yang sangat populer dalam dunia pengembangan perangkat lunak. Kedua bahasa ini digolongkan kedalam bahasa tingkat menengah. Semenjak dikembangkan, bahasa C dan C++ banyak digunakan untuk mengembangkan program-program aplikasi di bidang telekomunikasi *financial* atau bisnis dan sistem operasi. Bahkan sampai saat ini, pembuatan program-program untuk permainan komputer (*game*) sebagian besar masih menggunakan bahasa C/C++.⁷

1. Struktur Program C++

Struktur Program C++ terdiri dari sejumlah blok fungsi, setiap fungsi terdiri dari satu atau beberapa pernyataan yang berfungsi melaksanakan tugas tertentu.⁸

Contohnya sebagai berikut:

Gambar 3. Struktur Program C++



⁷ Budi Raharjo, *Pemrograman C++*. (Cet. 3; Bandung: Informatika Bandung), h. 4.

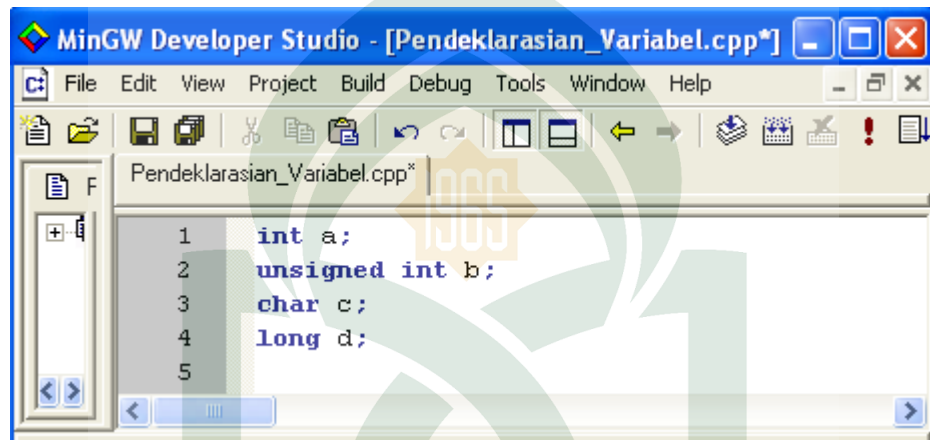
⁸ Lab Score. "Struktur Program C++". [http://ilmukamu.wordpress.com/2011/11/28/tipe-data-pada-c/\(diakses \(22 Februari 2012\)](http://ilmukamu.wordpress.com/2011/11/28/tipe-data-pada-c/(diakses%2022%20Februari%202012))

2. Tipe Data

Ada beberapa tipe data standar yang digunakan untuk data bilangan bulat yaitu int yang jangkauan nilainya -32768 s/d 32767, unsigned int jangkauan nilainya 0 s/d 65535, char jangkauan nilainya -128 s/d 127.

Cara pendeklarasian tipe data ini sama seperti pendeklarasian lainnya, yaitu :⁹

Gambar 4. Pendeklarasian Variabel



3. Variabel

Variabel adalah karakteristik yang dapat diamati dari suatu (objek), dan mampu memberikan bermacam-macam nilai atau beberapa kategori.¹⁰

Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan. Untuk memperoleh nilai dari suatu variable

⁹ Ilmu Kamu. "Tipe Data pada C++". <http://ilmukamu.wordpress.com/2/tipe-data-pada-c/>. (22 Februari 2012).

¹⁰ Riduwan, Akdon, *Rummus dan Data dalam Analisis Statistika* (cet, 1 ; Bandung : ALFABETA) h. 6

digunakan pernyataan penugasan (assignment statement), yang mempunyai sintaks sebagai berikut :¹¹

variable = ekspresi ;

Nama dari suatu variabel dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

- a. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C ++ bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara nim, NIM dan Nim dianggap berbeda.
 - b. Tidak boleh mengandung spasi.
 - c. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (*underscore*). Yang termasuk simbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, *, (,), -, +, = dsb.
 - d. Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.
4. Pernyataan kondisi

Pernyataan adalah kalimat yang memiliki nilai benar saja atau salah saja tetapi tidak sekaligus benar dan salah. Pernyataan kondisi ini terdiri dari IF, dan IF..Else, dan Switch.¹²

¹¹ Ilmu Kamu. “*Tipe Data pada C++*”.<http://ve511z.wordpress.com/2009/10/10/variabel-dalam-c/>. (22 Februari 2012).

¹² Muhammad Dinar. *Pengantar Dasar Matematika* (cet, 1 ; Makassar : FMIPA UNM).hal. 1.

D. Bahasa Pemrograman *Gentee*

Bahasa *Gentee* dikembangkan oleh *Alexi Krivonogov* pada tahun 2004. Bahasa pemrograman *Gentee* merupakan bahasa pemrograman prosedural tingkat tinggi. Sintaks dari bahasa pemrograman ini umumnya mempunyai kemiripan dengan sintaks C/C++. Hal ini dapat mempermudah pemakai menguasai banyak fitur secara cepat dalam menulis program pada bahasa *Gentee* ini. Sama dengan bahasa pemrograman *Java* atau C#, kode program *dicompile* ke dalam bentuk kode, setelah itu dieksekusi oleh mesin.¹³

1. Keuntungan *Gentee*

Keuntungan dari *Gentee* adalah suatu compiler, seperti suatu mesin virtual, yang dapat digunakan dalam aplikasi. Sebagai contoh, banyak orang menggunakan ungkapan reguler didalam program. Bagaimanapun, ini dapat mengambil keuntungan berharga dari bahasa pemrograman. Dapat menulis sendiri modul secara terpisah pada bahasa pemrograman *Gentee*. Jadi dengan kata lain bahasa pemrograman *Gentee* sama dengan bahasa pemrograman yang lainnya, sehingga penting untuk dipelajari secara mendalam. Yang pada umumnya harus membuat banyak rutin-rutin berbeda. Yang perlu dicatat bahwa, *Gentee* cukup sempurna untuk melakukan proses otomasi. Tidak harus memerlukan *compiler* mahal dan yang besar ukurannya. Hanya perlu menginstal *Gentee* dan membangun fungsi-fungsi yang akan benar-benar menyediakan banyak fitur-fitur dibandingkan batch file. Dan dapat mengeksekusi program didalam *Gentee* dan tidak harus menyimpan kode

¹³ Gentee, "*Gentee Overview Introduction*," <http://www.gentee.com> (02 Januari 2012).

“*compiler-produced*”, hanya cukup mengedit suatu teks dan menjalankan programnya.¹⁴

2. Struktur Program *Gentee*

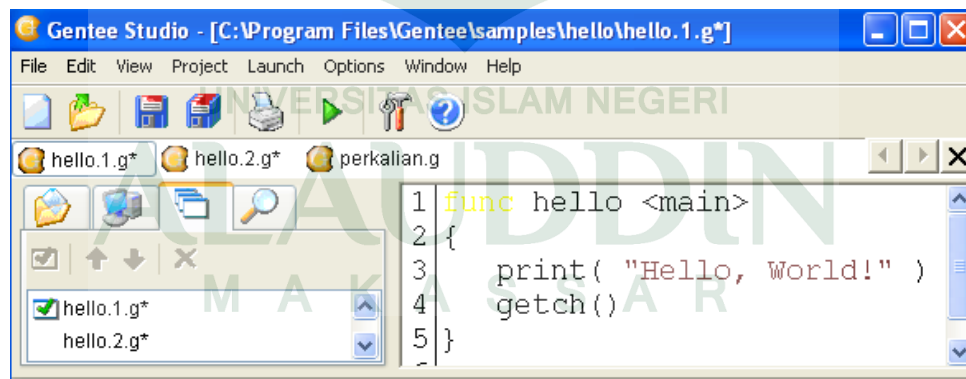
Program *Gentee* boleh terdiri dari satu atau beberapa file. Program *Gentee* mempunyai suatu unsur perintah utama. Suatu perintah dimulai dalam suatu baris baru, kebanyakan perintah berada didalam tanda kurung ‘{ }’

<program> ::= <command> {<end-of-line><command>} perintah-perintah :
function, method, operator, teks.

Bentuk perintah di atas dianggap sebagai suatu fungsi body. Bentuk perulangan dan bentuk statement mungkin tercakup dalam fungsi tersebut. Badan fungsi dan blok perulangan dianggap sebagai fungsi *blocks*. Sebagai tambahan, bentuk fungsi meliputi ekspresi, statement dan pendeklarasian variabel lokal.

Berikut contoh penulisan program *hello world* dalam bahasa *Gentee*:

Gambar 5. Program *Hello World*



¹⁴ Ibid., h. 1

Komentar dalam bahasa Gentee ada beberapa macam, program tidak akan memproses suatu perintah jika perintah tersebut diberikan tanda komentar.

Komentar-komentar dalam bahasa Gentee:

- `/*...*/`, adalah komentar secara perblok.
- `//`, adalah komentar perbaris.
- `;`, seperti tanda pemisah antar perintah.
- `:`, biasanya digunakan dalam seleksi kondisi (if,for,dll).

3. Tipe Data

Tipe data *uint* (*unsigned integer*), merupakan salah satu tipe data *integer*, dimana kombinasi dengan semua tipe struktur dan diberlakukan untuk menyimpan suatu alamat. Bahasa Gentee tidak mempunyai jenis tipe data *pointer*, tipe data *uint* yang digunakan. Untuk operasi perhitungan tipe data yang digunakan adalah :

sbyte, byte, short, ushort diperlakukan sebagai jenis int atau uint.

Tipe data *reserved* adalah tipe data khusus, dimana bukan menjadi anggota tipe data pokok ataupun struktur manapun. Tipe data ini ditandai oleh array bytes, dimana digambarkan dan digunakan sebagai suatu *array*. Yang membedakan tipe *reserved* adalah ruang memori yang dipesan, dimana tempat itu telah didefinisikan.

4. Deklarasi *variabel*

Variabel dideklarasikan dimanapun, di dalam suatu badan fungsi seperti

halnya di dalam perulangan blok. Masing-masing variabel harus diberi deklarasi tipe sendiri dalam baris baru, yang berisi nama variabel dan tipe data yang dipisahkan oleh tanda koma.

5. Operasi Elemen Array

Mayoritas objek dan struktur dapat berisi jenis struktur lainnya. Kurung siku '[']' digunakan untuk mendapatkan dan nilai dari item objek (unsur-unsur array, string karakter). Didalam kurung siku terdapat ekspresi jenis *uint*, itu yang harus terlampir didalam kurung siku. Jika suatu objek multidimensi digunakan, dimensinya dipisahkan oleh tanda koma. Indeks dimulai dari 0.

6. Statement Pengambilan Keputusan

a. *if – elif - else*

Statement ini terdiri dari beberapa bagian [5] :

- Bagian *if* berisi, suatu ekspresi kondisi dan blok eksekusi jika kondisi adalah *TRUE*. Jika kondisi *FALSE*, program akan melewatkannya lalu meneruskan ke bagian *elif*.
- Bagian *elif* berisikan, suatu ekspresi kondisi dan blok eksekusi jika kondisi adalah *TRUE*. Suatu statement mungkin berisi beberapa bagian *elif* yang berpindah dari satu demi satu.
- Bagian *else* berisikan, blok eksekusi jika kondisi dari *if* seperti semua kondisi *elif* adalah *FALSE*.

Operator *elif* dan *else* bersifat opsional. Nilai dari ekspresi kondisi harus numerik.

Nilai bernilai *TRUE* jika tidak bernilai nol.

7. Switch

Statement *switch* memungkinkan mengeksekusi satu statement atau lebih. Hal ini adalah subyek kepada nilai spesifik ekspresi. Bentuk ini digunakan untuk menyeleksi kondisi yang memiliki banyak kemungkinan (*multiple condition*), sama seperti multiple *if .. else*.

Statement *case* diikuti dengan ekspresi dipisahkan dengan tanda koma dan badan statement *case*. Statement ini dieksekusi bila ada ekspresi yang sama dengan nilai *switch*. Kontrol transfer dikerjakan hingga statement *switch* selesai. Ekspresi ditempatkan di dalam statement *case*, dievaluasi dan dibandingkan untuk persamaan secara sekuen. Jika semua ekspresi tidak sama dengan nilai *switch*, maka *control* akan *ditransfer* / dipindahkan ke case berikutnya atau ke statement default. Statement default tidak memiliki daftar ekspresi manapun; meskipun demikian, jika tidak satupun dari nilai-nilai *case* yang sesuai, blok *default* akan dieksekusi.

8. Statement Perulangan

a. While, Do-While

Statement *while* adalah statement perulangan yang simpel. Statement *while* mempunyai beberapa bagian : *keyword*, *ekspresi* kondisi dan badan perulangan (*blok*).

<while> ::= while <expression><block>

Eksekusi pada badan perulangan diulangi sampai terdapat nilai dari statement bernilai *FALSE*. Perulangan tidak akan pernah dieksekusi, bila bernilai nol pada saat test dilakukan pertama kalinya.

Statement *do* memiliki kata kunci *do*, suatu badan perulangan, kata kunci *while* dan ekspresi kondisi.

<dowhile> ::= do <block> while <expression>

Eksekusi badan perulangan selalu diulangi sampai nilai dari ekspresi bernilai *FALSE*. Tidak seperti statemen *while*, perulangan *do-while* ini paling tidak menjalankan perintahnya sebanyak satu kali.

9. *For* dan *Forum*

Statement *for* berisi kata kunci *for*, suatu urutan tiga ekspresi yang dipisahkan oleh tanda koma, suatu badan perulangan.

<for> ::= for [<expression>] ',' <expression> ','
[<expression>] <block>

digunakan untuk menambahkan atau mengurangi suatu nilai.

Jika perulangan adalah penjumlahan dengan nilai satu dan nilai tertinggi dari penjumlahan tersebut didefinisikan sebelum iterasi perulangan dimulai, statement *forum* digunakan didalam tempat statement *for*.

<forum> ::= forum <variable name> ['=<expression>]
'<expression> <block>

Kata kunci *forum* diikuti oleh suatu variabel *counter*, kemudian operator penugasan dan ekspresi (nilai awal dari *counter*) dapat digunakan. Jika

tidak terdapat operator penugasan, nilai inisialisasi counter tidak akan berubah. Bilangan bulat manapun harus diperlakukan sebagai suatu *variabel counter*. Ekspresi ini dievaluasi satu kali, sebelum iterasi perulangan dimulai. Perulangan dijalankan selama nilai dari *counter* lebih kecil dari nilai suatu ekspresi. Kemudian badan perulangan mengikuti. Secara *default* operasi penaikan (nilai dari *counter* ditambah dengan satu) bertambah.

E. Kriteria Perbandingan Bahasa Program

Suatu bahasa pemrograman yang baik adalah bahasa yang dapat dimengerti oleh manusia dengan mudah dan juga dapat dieksekusi dengan mudah oleh mesin. Namun dalam kenyataannya, prinsip tersebut tidak dapat diterapkan karena perbedaan antara bahasa mesin dan bahasa manusia. Karena itu dilakukan pendekatan-pendekatan linguistik oleh manusia sehingga manusia dapat secara tidak langsung “memerintah” mesin untuk berbagai keperluan. Perkembangan teknologi menyebabkan berkembangnya berbagai macam bahasa pemrograman yang bervariasi dan betebaran. Untuk itu riset dilakukan untuk melihat dan menganalisa bahasa mana yang dapat dikatakan terbaik. Untuk mendapatkan hasil maksimal riset ini dibatasi pada beberapa kriteria dasar yaitu:

1. Besar kecilnya program hasil kompilasi, program yang dibuat dengan bahasa pemrograman tertentu akan dikompilasi menjadi kode-kode yang dapat dimengerti oleh mesin. Besar-kecilnya ukuran program dapat menentukan baik-tidaknya program menyangkut efisiensi penggunaan memori.

2. Kecepatan program, program dapat dikatakan baik apabila dapat mengeksekusi instruksi-instruksi yang diberikan secara cepat.¹⁵

F. Perkalian Matriks

Matriks adalah jajaran elemen (dalam hal ini elemennya berupa bilangan) berbentuk empat persegi panjang. Bentuk (ukuran) matriks ditentukan oleh banyaknya baris dan kolom. Matriks yang mempunyai m baris dan n kolom dituliskan sebagai matriks berukuran $m \times n$, dan ukuran ini biasa dinamakan dengan ordo matriks.¹⁶

Dengan representasi matriks, perhitungan dapat dilakukan dengan lebih terstruktur. Pemanfaatannya misalnya dalam menjelaskan persamaan linier, transformasi koordinat, dan lainnya. Matriks seperti halnya variabel biasa dapat dimanipulasi, seperti dikalikan, dijumlah, dikurangkan dan didekomposisikan.

Gambar 6. Matriks

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

¹⁵ Satria, "Perbandingan Bahasa Pemrograman (C, C++, D, Basic, Pascal)". <http://satriaskyterror.wordpress.com/2010/10/08/perbandingan-bahasa-pemrograman/> (22 Februari 2012).

¹⁶ Heri Purwanto, Gina Indriani, Erlina Dayanti, *Aljabar Linier* (Cet, 1; Jakarta: PT. Ercobntara Rajawali, 2005), h.21.

Suatu matriks biasanya dinotasikan dengan huruf kapital ditebalkan (misal matriks A, dituliskan dengan **A**). jika A adalah matriks dengan ukuran $m \times r$ dan B adalah matriks ukuran $r \times n$, maka hasil kali AB adalah matriks $m \times n$. Dengan kata lain dua matriks ini dapat dikalikan jika ukuran kolom matriks A sama dengan baris matriks B.¹⁷

Adapun Sifat-sifat perkalian matriks :

Jika diketahui A, B dan C suatu matriks, maka perkalian matriks memenuhi sifat-sifat:

a. AB dan BA (Tidak selalu Komutatif)

AB dan BA tidak selalu setara. Kesetaraan tidak terjadi karena tiga alasan: Kemungkinan pertama, hasil kali AB dapat didefinisikan tetapi BA tidak dapat didefinisikan. Sebagai contoh,

Jika $A = \begin{bmatrix} 2 & 3 \\ -4 & 6 \end{bmatrix}$ dan $B = \begin{bmatrix} 5 & 0 & 3 \\ 2 & 4 & -1 \end{bmatrix}$ maka,

$$AB = \begin{bmatrix} 2 & 3 \\ -4 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 0 & 3 \\ 2 & 4 & -1 \end{bmatrix} = \begin{bmatrix} 16 & 12 & 3 \\ -8 & 24 & -18 \end{bmatrix}$$

Sedangkan $BA = \begin{bmatrix} 5 & 0 & 3 \\ 2 & 4 & -1 \end{bmatrix} \times \begin{bmatrix} 2 & 3 \\ -4 & 6 \end{bmatrix}$ = tidak ada hasil kalinya, karena $A_{2 \times 2}$ dan $B_{2 \times 3}$.

Mungkin pula terjadi bahwa AB dan BA keduanya dapat didefinisikan tetapi memiliki ukuran yang berbeda. Ini terjadi jika A adalah matriks 2×3

¹⁷ Muhammad Arhami, Anita Desiani, *Pemrograman MATLAB* (cet, 1 ; Yogyakarta : ANDI) h. 73.

dan B adalah matriks 3x2. Kemungkinan terakhir, sebagaimana pada contoh berikut:

$$\text{Misal : } A = \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

Dengan mengalikan keduanya diperoleh,

$$AB = \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -2 \\ 11 & 4 \end{bmatrix}$$

$$BA = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ -3 & 0 \end{bmatrix}$$

Diperoleh $AB \neq BA$, meskipun AB dan BA dapat didefinisikan dan memiliki ukuran yang sama.

b. $A(BC) = (AB)C$ (Asosiatif)

Jika $A = [a_{ij}]_{m \times n}$, $B = [b_{ij}]_{n \times q}$ dan $C = [c_{ij}]_{q \times r}$ maka,

Ruas kiri :

$$BC = [x_{ij}]_{n \times r}, \text{ dengan } x_{ij} = b_{i1}c_{1j} + b_{i2}c_{2j} + \dots + b_{iq}c_{qj} = \sum_{k=1}^q b_{ik}c_{kj}$$

$$A(BC) = [y_{ij}]_{m \times r}, \text{ dengan } y_{ij} = a_{i1}x_{1j} + a_{i2}x_{2j} + \dots + a_{in}x_{nj}$$

$$\begin{aligned} &= a_{i1}(b_{11}c_{1j} + b_{12}c_{2j} + \dots + b_{1q}c_{qj}) + a_{i2}(b_{21}c_{1j} + b_{22}c_{2j} + \dots + b_{2q}c_{qj}) \\ &\quad + \dots + a_{in}(b_{n1}c_{1j} + b_{n2}c_{2j} + \dots + b_{nq}c_{qj}). \\ &= \sum_{k=1}^q a_{i1}b_{1k}c_{kj} + \sum_{k=1}^q a_{i2}b_{2k}c_{kj} + \dots + \sum_{k=1}^q a_{in}b_{nk}c_{kj} \\ &= \sum_{l=1}^n \sum_{k=1}^q a_{il}b_{lk}c_{kj} \end{aligned}$$

Ruas Kanan :

$$AB = [z_{ij}]_{m \times q}, \text{ dengan } z_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{k=1}^n a_{ik}b_{kj}$$

$$\begin{aligned}
(AB)C &= [w_{ij}]_{m \times r}, \text{ dengan } w_{ij} = a_{i1}z_{1j} + a_{i2}z_{2j} + \dots + a_{iq}z_{qj} \\
&= (a_{i1}b_{11} + a_{12}b_{2j} + \dots + a_{in}b_{n1})c_{1j} + (a_{i1}b_{12} + a_{12}b_{22} + \dots + a_{in}b_{n2})c_{2j} \\
&\quad + \dots + (a_{i1}b_{1q} + a_{12}b_{2q} + \dots + a_{in}b_{nq})c_{qj} \\
&= \sum_{l=1}^n a_{il}b_{l1}c_{1j} + \sum_{l=1}^n a_{il}b_{l2}c_{2j} + \dots + \sum_{l=1}^n a_{il}b_{lq}c_{qj} \\
&= \sum_{l=1}^n \sum_{k=1}^q a_{il}b_{lk}c_{kj}
\end{aligned}$$

Jadi $A(BC) = (AB)C$ terbukti.

c. $A(B+C) = AB + AC$ (Distributif)

Bukti :

Jika $A = [a_{ij}]_{m \times n}$, $B = [b_{ij}]_{n \times q}$ dan $C = [c_{ij}]_{n \times q}$ maka,

Ruas kiri :

$$(B + C) = [b_{ij} + c_{ij}]_{n \times q}$$

$$A(B + C) = [x_{ij}]_{m \times q}, \text{ dengan } x_{ij} = \sum_{k=1}^n a_{ik}(b_{kj} + c_{kj})$$

Ruas kanan :

$$AB = [y_{ij}]_{m \times q}, \text{ dengan } y_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$$

$$AC = [z_{ij}]_{m \times q}, \text{ dengan } z_{ij} = \sum_{k=1}^n a_{ik}c_{kj}$$

$$AB + AC = \sum_{k=1}^n a_{ik}b_{kj} + \sum_{k=1}^n a_{ik}c_{kj}$$

$$= \sum_{k=1}^n a_{ik}(b_{kj} + c_{kj})$$

$$= \sum_{k=1}^n a_{ik}(b_{kj} + c_{kj})$$

Jadi $A(B+C) = AB + AC$ terbukti.

Adapun Syarat Perkalian Dua Matriks :

Jika matriks $A_{m \times n}$ dan matriks $B_{p \times q}$ dikalikan, maka :

1. Banyaknya kolom matriks A harus sama dengan banyaknya kolom matriks B, sehingga $n = p$
2. Matriks hasil perkalian antara A dan B adalah matriks dengan ordo $m \times q$
3. Perkalian dilakukan dengan menjumlahkan hasil kali setiap elemen baris matriks A dengan setiap elemen kolom matriks B yang sesuai.

Contoh perkalian matriks :

Misalkan matriks A yang memiliki baris 2 dan kolom 3, matriks B yang memiliki baris 3 dan kolom 2.

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 0 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & -4 \\ 1 & 5 \\ -2 & 2 \end{bmatrix}$$
$$A \times B = \begin{bmatrix} 1(3) + 2(1) + 1(-2) & 1(-4) + 2(5) + 1(2) \\ 4(3) + 0(1) + 2(-2) & 4(-4) + 0(5) + 2(2) \end{bmatrix}$$
$$= \begin{bmatrix} 3 & 8 \\ 8 & -12 \end{bmatrix}$$

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian terapan. Penelitian terapan adalah salah satu jenis penelitian yang bertujuan untuk memberikan solusi atas permasalahan tertentu secara praktis. Penelitian ini tidak berfokus pada pengembangan sebuah ide, teori atau gagasan, tetapi lebih berfokus pada aplikasi dari penelitian yang akan diterapkan pada perkalian matriks.

B. Sumber Data

Sumber data dalam penelitian dilakukan dengan studi pustaka mengenai buku yang membahas bahasa pemrograman. Dimana studi pustaka dilakukan penulis agar mendapatkan data yang bersifat teoritis. Dalam hal ini yang perlu diperhatikan objek yang diteliti, khususnya buku bacaan yang dijadikan pedoman untuk pembahasan masalah yang didapat dari penelitian yang dilakukan.

C. Lokasi Penelitian

Adapun lokasi penelitian adalah di Perpustakaan UIN Alauddin Makassar yang menyediakan buku-buku penunjang mengenai buku bahasa pemrograman Gentee, bahasa pemrograman C++ dan Perkalian Matriks.

D. Prosedur Penelitian

Adapun langkah-langkah dalam penelitian ini adalah :

1. Membuat aplikasi perkalian matriks dengan bahasa pemrograman Gentee dan bahasa pemrograman C++. Adapun langkah-langkahnya adalah:
 1. Melakukan pengumpulan data untuk pembuatan program bahasa Gentee dan bahasa C++.
 2. Menyusun algoritma dan membuat flowchart yang dapat dijalankan secara sama dalam bahasa Gentee ataupun C++.
 3. Menyiapkan sarana pendukung perangkat keras yang identik dalam pengujian ini. Pengujian akan dilakukan pada satu mesin yang sama.
 4. Melakukan proses coding pada program bahasa Gentee dan bahasa C++.
 5. Menjalankan aplikasi yang dibuat, pada percobaan ini dilakukan pada aplikasi perkalian matriks. Melihat ada pesan error yang terjadi atau tidak.
2. Membandingkan bahasa pemrograman Gentee dan bahasa pemrograman C++ dengan pembuatan aplikasi yang sama. Adapun kriterianya yaitu :
 1. Membandingkan penggunaan statement-statement dan *syntax-syntax* yang digunakan dalam bahasa Pemrograman Gentee dengan bahasa Pemrograman C++.
 2. Membandingkan kecepatan eksekusi bahasa Gentee dan bahasa C++ pada saat program dijalankan.

3. Membandingkan ukuran penggunaan memori pada saat program dibuat menggunakan bahasa Gntee dan bahasa C++ melalui mesin penyimpanan yang sama.



BAB IV

HASIL DAN PEMBAHASAN

A. Hasil

Berdasarkan pada teori yang telah di bahas pada bab-bab sebelumnya, dalam bab ini akan dibahas algoritma, flowchart, dan program aplikasi perkalian matriks dengan menggunakan bahasa perograman Gentee dan bahasa pemrograman C++.

1. Algoritma Perkalian Matriks

Adapun algoritma perkalian matriks dalam pembuatan program yaitu :

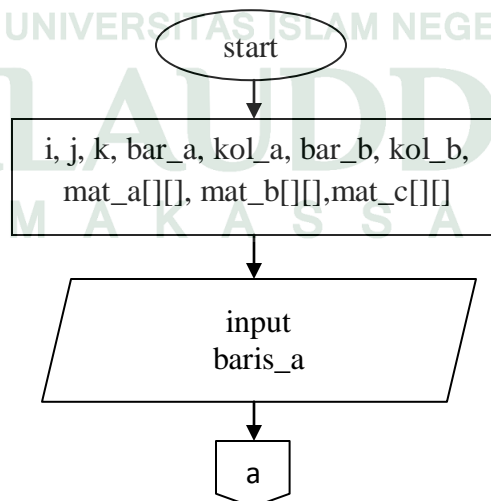
- a. Mulai.
- b. Deklarasikan variabel i, j, k, bar_a, kol_a_dan_bar_b, kol_b, mat_a[][], mat_b[][], mat_c[][].
- c. Masukkan nilai bar_a
- d. Masukkan nilai kol_a_dan_bar_b
- e. Masukkan nilai kol_b
- f.
 - Jika nilai bar_a sama dengan nilai kolom_b, lanjut ke g.
 - Jika tidak kembali ke d.
- g.
 - Inisialisasi i = 0 dan i < bar_a,
 - Inisialisasi j = 0 dan j < kol_a_dan_bar_b.
 - Masukkan nilai mat_a[i][j].
 - Cetak mat_a[i][j].
- h.
 - Inisialisasi i = 0 dan i < kol_a_dan_bar_b,

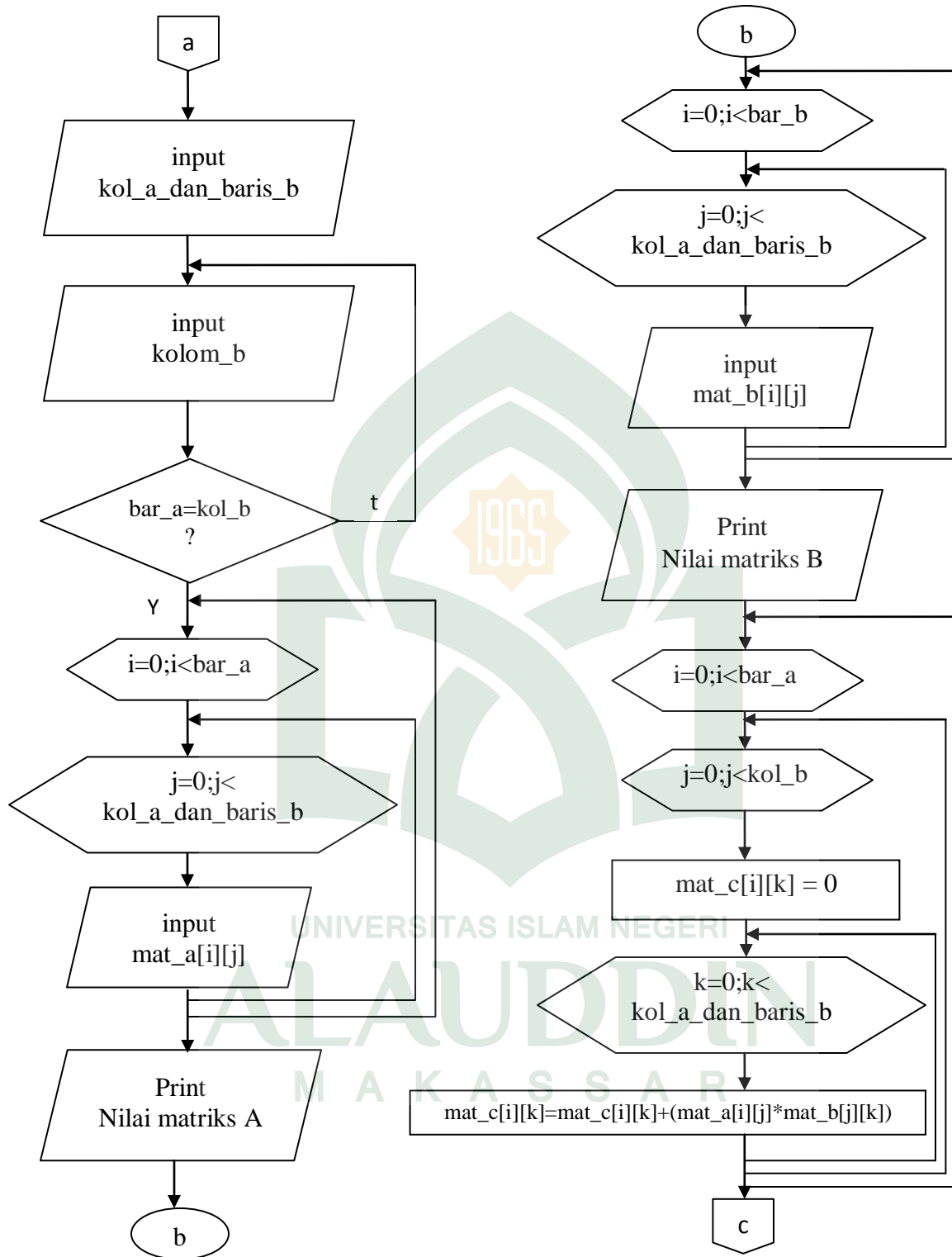
- Inisialisasi $j = 0$ dan $j < \text{kol_b}$.
- Masukkan nilai $\text{mat_b}[i][j]$.
- Cetak $\text{mat_b}[i][j]$.
- i. - Inisialisasi $i = 0$ dan $i < \text{bar_a}$,
 - Inisialisasi $j = 0$ dan $j < \text{kol_b}$.
 - Inisialisasi $\text{mat_c}[i][j] = 0$
 - Inisialisasi $k = 0$ dan $k < \text{kol_a_dan_bar_b}$
 - Jumlahkan $\text{mat_c}[i][j] = \text{mat_c}[i][j] + \text{mat_a}[i][k] \times \text{mat_b}[k][j]$
- j. - Inisialisasi $i = 0$ dan $i < \text{bar_a}$,
 - Inisialisasi $j = 0$ dan $j < \text{kol_b}$.
 - Cetak $\text{mat_c}[i][j]$
- k. Selesai

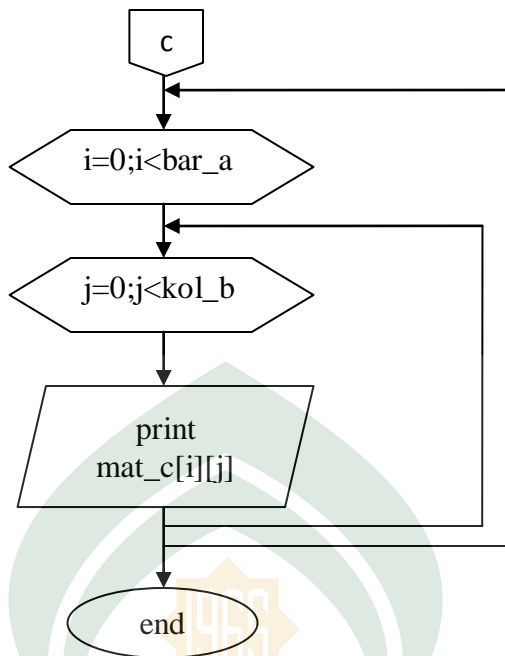
2. Flowchart Perkalian Matriks

Adapun flowchart perkalian matriks dalam pembuatan program yaitu :

Gambar 7. Flowchart Perkalian Matriks







4. Penulisan code

Penulisan kode yaitu dengan memasukkan kode-kode *program* ke dalam bahasa *program* yang kemudian akan memberikan hasil atau *output* yang diinginkan. Adapun penulisan kode pada penelitian ini yaitu untuk bahasa pemrograman C++ menggunakan *software MinGW Developer Studio* dan untuk bahasa pemrograman Gentee menggunakan *software Gentee Studio*.

Pembuatan aplikasi perkalian matriks pemrograman C++ menggunakan *software MinGW Developer Studio* yaitu :

Gambar 8. Aplikasi Perkalian Matriks C++

```

1  #include<iostream>
2  #include<iomanip>
3  using namespace std;
4  int main() {
5  int i, j, k, bar_a, kol_a_dan_bar_b, kol_b, mat_a[100][100], mat_b[100][100], mat_c[100][100];

```

```

6 | cout<<"Masukkan Jumlah Baris Matriks A"<<endl;
7 | cout<<"====="<<endl;
8 | cout<<"Jumlah baris = ";
9 | cin>>bar_a;
10 | cout<<endl;
11 | cout<<"Masukkan Jumlah Kolom Matriks A Dan Baris Matriks B"<<endl;
12 | cout<<"====="<<endl;
13 | cout<<"jumlah kol_A dan bar_B = ";
14 | cin>>kol_a_dan_bar_b;
15 | cout<<endl;
16 | langsung;
17 | cout<<"Masukkan Jumlah Kolom Matriks B"<<endl;
18 | cout<<"====="<<endl;
19 | cout<<"jumlah kolom = ";
20 | cin>>kol_b;
21 | cout<<endl;
22 | cout<<endl;
23 | if(bar_a!=kol_b){
24 |     goto langsung;
25 | }
26 |     cout<<"Matriks A "<<bar_a<<" X "<<kol_a_dan_bar_b<<endl;
27 |     cout<<"====="<<endl;
28 |     for(i=0;i<bar_a;i++){
29 |         for(j=0;j<kol_a_dan_bar_b;j++){
30 |             cout<<"Masukkan nilai "<<i+1<<","<<j+1<<"= ";
31 |             cin>>mat_a[i][j];
32 |         }
33 |     }
34 |     cout<<endl;
35 |     cout<<"Jadi Matriks A "<<bar_a<<" X "<<kol_a_dan_bar_b<<endl;
36 |     cout<<"====="<<endl;
37 |     cout<<endl;
38 |     for(i=0;i<bar_a;i++){
39 |         for(j=0;j<kol_a_dan_bar_b;j++){
40 |             cout<<setw(4)<<mat_a[i][j];
41 |         }
42 |         cout<<endl;
43 |     }
44 |     cout<<endl;
45 |     cout<<endl;
46 |     cout<<"Matriks B "<<kol_a_dan_bar_b<<" X "<<kol_b<<endl;
47 |     cout<<"====="<<endl;
48 |     for(i=0;i<kol_a_dan_bar_b;i++){
49 |         for(j=0;j<kol_b;j++){
50 |             cout<<"Masukkan nilai "<<i+1<<","<<j+1<<"= ";
51 |             cin>>mat_b[i][j];
52 |         }
53 |     }
54 |     cout<<endl;
55 |     cout<<endl;
56 |     cout<<"Jadi Matriks B "<<kol_a_dan_bar_b<<" X "<<kol_b<<endl;
57 |     cout<<"====="<<endl;
58 |     cout<<endl;
59 |     for(i=0;i<kol_a_dan_bar_b;i++){
60 |         for(j=0;j<kol_b;j++){
61 |             cout<<setw(4)<<mat_b[i][j];
62 |         }
63 |         cout<<endl;
64 |     }
65 |     //perkalian matriks A x B
66 |     for(i=0;i<bar_a;i++){
67 |         for(j=0;j<kol_b;j++){

```

```

68     mat_c[i][j]=0;
69     for(k=0;k<kol_a_dan_bar_b;k++){
70         mat_c[i][j]=mat_c[i][j] + mat_a[i][k]*mat_b[k][j];
71     }
72 }
73
74 //perkalian matriks B x A
75 for(i=0;i<kol_b;i++){
76     for(j=0;j<bar_a;j++){
77         mat_c[i][j]=0;
78         for(k=0;k<kol_a_dan_bar_b;k++){
79             mat_c[i][j]=mat_c[i][j] + mat_b[i][k]*mat_a[k][j];
80         }
81     }
82 }
83 cout<<endl;
84 cout<<endl;
85 cout<<"Jadi Matriks A X B"<<endl;
86 cout<<"====="<<endl;
87 cout<<endl;
88 for(i=0;i<bar_a;i++){
89     for(j=0;j<kol_b;j++){
90         cout<<setw(4)<<mat_c[i][j];
91     }
92     cout<<endl;
93 }
94 cout<<endl;
95 cout<<endl;
96 cout<<"Jadi Matriks B X A"<<endl;
97 cout<<"====="<<endl;
98 cout<<endl;
99 for(i=0;i<kol_b;i++){
100     for(j=0;j<bar_a;j++){
101         cout<<setw(4)<<mat_c[i][j];
102     }
103     cout<<endl;
104 }
105 return 0;
106 }

```

Penjelasan Program di atas :

Dalam program C++ sebagai awal program digunakan `#include` yang merupakan salah satu pengarah preprocessor directive yang merupakan penginstruksi kepada compiler yang menyisipkan file yang tersedia pada C++. `Main()`, merupakan fungsi utama dimana seluruh program akan dieksekusi, diletakkan di awal program, isi dari fungsi main akan selalu dieksekusi pertama kali. Main diikuti dengan sepasang tanda kurung `()` karena merupakan fungsi, isi dari fungsi main selanjutnya

berupa deklarasi formal dan dituliskan diantara kurung kurawal {}. Pada program di atas variable yang digunakan semua bertipe data integer, kemudian dilanjutkan dengan perintah menginput nilai baris dan kolom menggunakan perintah cout dan cin, kemudian untuk menginput nilai indeks digunakan proses perulangan yaitu for bersarang sehingga proses penginputan nilai akan terus berulang hingga terpenuhi syarat yang ada. Begitu pula proses mencetak matriks dan hasil perkaliannya. Pada saat proses penginputan nilai baris dan kolom matriks A dan B selesai maka akan dilakukan pemeriksaan jika nilai kolom matriks A sama dengan nilai baris B maka proses penginputan nilai-nilai matriks A dan B akan dilakukan kemudian mencetak dan melakukan proses perkalian serta menampilkan hasil perkalian matriks A dan B. Jika syarat tidak terpenuhi maka program akan mencetak bahwa nilai kolom matriks A dan nilai baris matriks B tidak sama. Setiap *statement* yang ada pada program C++ diakhiri dengan tanda titik koma (;).

- Pembuatan aplikasi perkalian matriks pemrograman Gentee menggunakan *software Gentee Studio* yaitu :

Gambar 9. Aplikasi Perkalian Matriks Gentee

```

1 func main<main>
2 {
3     str input
4     int i j k bar_a kol_a dan_bar_b kol_b langsung
5     arr mat_a[100,100]
6     arr mat_b[100,100]
7     arr mat_c[100,100]
8     print("\nMasukkan Jumlah Baris Matriks A\n")
9     print("=====\n")
10    print("Jumlah baris = ")

```



```

11 | bar_a = int(conread(input))
12 | print("\n")
13 | print("\n")
14 | print("Masukkan Jumlah Kolom Matriks A Dan Baris Matriks B\n")
15 | print("=====\n")
16 | print("jumlah kol_A dan bar_B = ")
17 | kol_a_dan_bar_b = int(conread(input))
18 | print("\n")
19 | print("\n")
20 | label langsung
21 | print("Masukkan Jumlah Kolom Matriks B\n")
22 | print("=====\n")
23 | print("Jumlah kolom = ")
24 | kol_b = int(conread(input))
25 | print("\n")
26 | print("\n")
27 | if bar_a!=kol_b {
28 |     goto langsung
29 | }
30 | print("\nMatriks A \ (bar_a) x \ (kol_a_dan_bar_b)\n")
31 | print("=====\n")
32 | for i=0,i<bar_a,i++
33 | {
34 |     for j=0,j<kol_a_dan_bar_b,j++
35 |     {
36 |         print("Masukkan nilai \ (i+1) .\ (j+1) = ")
37 |         mat_a[i,j]=int(conread(input))
38 |     }
39 | }
40 | print("\n")
41 | print("\nJadi Matriks A \ (bar_a) x \ (kol_a_dan_bar_b)\n")
42 | print("=====\n")
43 | for i=0,i<bar_a,i++
44 | {
45 |     for j=0,j<kol_a_dan_bar_b,j++
46 |     {
47 |         print("\t\ (mat_a[i,j]) ")
48 |     }
49 |     print("\n")
50 | }
51 | print("\n")
52 | print("\n")
53 | print("\nMatriks B \ (kol_a_dan_bar_b) x \ (kol_b)\n")
54 | print("=====\n")
55 | for i=0,i<kol_a_dan_bar_b,i++
56 | {

```

```

57         for j=0,j<kol_b,j++
58         {
59             print("Masukkan nilai \ (i+1).\ (j+1) = ")
60             mat_b[i,j]=int(conread(input))
61         }
62     }
63     print("\n")
64     print("\nJadi Matriks B \ (kol_a_dan_bar_b) x \ (kol_b)\n")
65     print("=====\n")
66     for i=0,i<kol_a_dan_bar_b,i++
67     {
68         for j=0,j<kol_b,j++
69         {
70             print("\t\ (mat_b[i,j])")
71         }
72         print("\n")
73     }
74     for i=0,i<bar_a,i++
75     {
76         for j=0,j<kol_b,j++
77         {
78             mat_c[i,j]=0;
79             for k=0,k<kol_a_dan_bar_b,k++
80             {
81                 mat_c[i,j]=mat_c[i,j] + mat_a[i,k]*mat_b[k,j];
82             }
83         }
84     }
85     for i=0,i<kol_b,i++
86     {
87         for j=0,j<bar_a,j++
88         {
89             mat_c[i,j]=0;
90             for k=0,k<kol_a_dan_bar_b,k++
91             {
92                 mat_c[i,j]=mat_c[i,j] + mat_b[i,k]*mat_a[k,j];
93             }
94         }
95     }
96     print("\n")
97     print("\nJadi Matriks A X B\n")
98     print("=====\n")
99     for i=0,i<bar_a,i++
100    {
101        for j=0,j<kol_b,j++
102    {

```

```

103         print("\t\ (mat_c[i,j])")
104     }
105     print("\n")
106 }
107     print("\n")
108     print("\nJadi Matriks B X A\n")
109     print("=====\n")
110     for i=0,i<kol_b,i++
111     {
112         for j=0,j<bar_a,j++
113         {
114             print("\t\ (mat_c[i,j])")
115         }
116         print("\n")
117     }
118     getch()
119 }

```

Penjelasan Program di atas :

Awal program dimulai dengan fungsi utama kemudian melakukan inisialisai variabel yang bertipe data integer, array dan string, lalu melakukan perintah penginputan nilai baris dan kolom pada matriks A dan B dengan menggunakan perintah printf. Setelah melakukan perintah penginputan selanjutnya akan diperiksa jika baris kolom A dan baris matriks B sama maka akan dilakukan proses penginputan nilai-nilai matriks A dan B dengan menggunakan perintah for yang akan melakukan penginputan berulang-ulang hingga syarat terpenuhi. Begitu pula Untuk mencetak hasil perkalian matriks A dan B.

5. Spesifikasi Perangkat Keras yang digunakan

Perangkat keras (hardware) yang digunakan untuk pembuatan aplikasi perkalian matriks bahasa C++ dan bahasa Gentee adalah 1 unit komputer dengan spesifikasi :

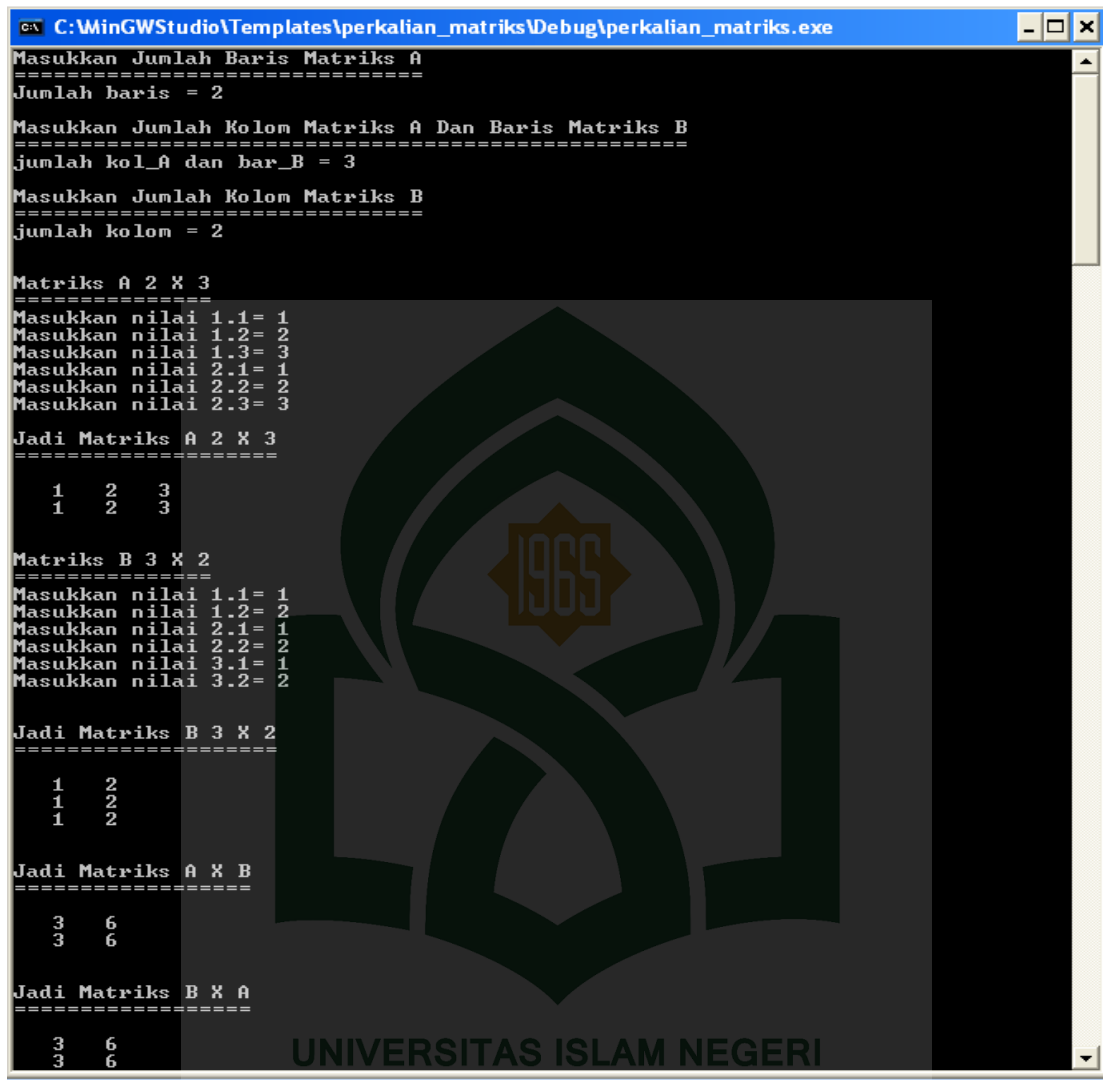
1. Processor Pentium Dual 1.73 Mhz
2. RAM 502 MB
3. VGA 128 MB

B. Pembahasan

Berdasarkan hasil pembuatan aplikasi yang telah dibahas pada bab sebelumnya, dapat dilakukan suatu analisa yaitu yang pertama, kita harus menjalankan kedua program yang ada, kemudian untuk mendapatkan hasil maksimal penelitian ini dibatasi pada beberapa kriteria dasar yaitu:

1. Besar kecilnya program hasil kompilasi, program yang dibuat dengan bahasa pemrograman tertentu akan dikompilasi menjadi kode-kode yang dapat dimengerti oleh mesin. Besar-kecilnya ukuran program dapat menentukan baik-tidaknya program menyangkut efisiensi penggunaan memori.
2. Kecepatan program, program dapat dikatakan baik apabila dapat mengeksekusi instruksi-instruksi yang diberikan secara cepat.
 - a. Eksekusi Program
 - Hasil eksekusi kode program perkalian matriks pada bahasa C++ yaitu :

Gambar 10. Output Aplikasi Perkalian Matriks C++



```
C:\WinGWStudio\Templates\perkalian_matriks\Debug\perkalian_matriks.exe
Masukkan Jumlah Baris Matriks A
=====
Jumlah baris = 2

Masukkan Jumlah Kolom Matriks A Dan Baris Matriks B
=====
jumlah kol_A dan bar_B = 3

Masukkan Jumlah Kolom Matriks B
=====
jumlah kolom = 2

Matriks A 2 X 3
=====
Masukkan nilai 1.1= 1
Masukkan nilai 1.2= 2
Masukkan nilai 1.3= 3
Masukkan nilai 2.1= 1
Masukkan nilai 2.2= 2
Masukkan nilai 2.3= 3

Jadi Matriks A 2 X 3
=====
1 2 3
1 2 3

Matriks B 3 X 2
=====
Masukkan nilai 1.1= 1
Masukkan nilai 1.2= 2
Masukkan nilai 2.1= 1
Masukkan nilai 2.2= 2
Masukkan nilai 3.1= 1
Masukkan nilai 3.2= 2

Jadi Matriks B 3 X 2
=====
1 2
1 2
1 2

Jadi Matriks A X B
=====
3 6
3 6

Jadi Matriks B X A
=====
3 6
3 6
```

Pada program perkalian matriks dengan menggunakan bahasa C++ digunakan array berdimensi dua untuk membuat matriks $n \times n$, nilai baris dan kolom matriks A akan di masukkan melalui *keyboard* begitupula dengan matriks B yang kemudian akan diperiksa oleh program jika memenuhi syarat bahwa nilai kolom pada matriks A sama dengan baris pada matriks B maka pengisian anggota matriks A dan B

dilakukan, kemudian program akan melakukan perkalian matriks A dan B dan akan menampilkan hasil perkalian dari matriks A dan B. Jika syarat yang tadi tidak terpenuhi maka program akan berhenti dan akan mencetak atau hasil outputnya sebagai berikut :

- Hasil eksekusi kode program perkalian matriks pada bahasa Gentee yaitu :

Gambar 11. Output Aplikasi Perkalian Matriks Gentee

```

C:\Program Files\Gentee\exe\gentee.exe
Masukkan Jumlah Baris Matriks A
=====
Jumlah baris = 2

Masukkan Jumlah Kolom Matriks A Dan Baris Matriks B
=====
jumlah kol_A dan bar_B = 3

Masukkan Jumlah Kolom Matriks B
=====
Jumlah kolom = 2

Matriks A 2 x 3
=====
Masukkan nilai 1.1 = 1
Masukkan nilai 1.2 = 2
Masukkan nilai 1.3 = 3
Masukkan nilai 2.1 = 1
Masukkan nilai 2.2 = 2
Masukkan nilai 2.3 = 3

Jadi Matriks A 2 x 3
=====
      1      2      3
1      1      2      3

Matriks B 3 x 2
=====
Masukkan nilai 1.1 = 1
Masukkan nilai 1.2 = 2
Masukkan nilai 2.1 = 1
Masukkan nilai 2.2 = 2
Masukkan nilai 3.1 = 1
Masukkan nilai 3.2 = 2

Jadi Matriks B 3 x 2
=====
      1      2
1      1      2
1      1      2

Jadi Matriks A x B
=====
      3      6
3      3      6

Jadi Matriks B x A
=====
      3      6
3      3      6
  
```

Pada program perkalian matriks dengan menggunakan bahasa Gentee digunakan juga array berdimensi dua untuk membuat matriks $n \times n$, nilai baris dan kolom matriks A akan di masukkan melalui *keyboard* begitupula dengan matriks B yang kemudian akan diperiksa oleh program jika memenuhi syarat bahwa nilai kolom pada matriks A sama dengan baris pada matriks B maka pengisian anggota matriks A dan B dilakukan, kemudian program akan melakukan perkalian matriks A dan B dan akan menampilkan hasil perkalian dari matriks A dan B.

b. Memori dan kecepatan Program

- Dari hasil pembuatan program aplikasi perkalian matriks, memori program dengan menggunakan bahasa program c++ yaitu 540, memori program dengan menggunakan bahasa program Gentee yaitu 3 kb.
- Dari Hasil pembuatan program aplikasi perkalian matriks, kecepatan program yang didapatkan setelah melakukan 10 kali yaitu :

Tabel 2. Hasil Eksekusi Program Aplikasi Perkalian Matriks

No.	C++	Gentee
1	0.0313	0.0265
2	0.0334	0.0235
3	0.0311	0.0229
4	0.0343	0.0216
5	0.0336	0.0224
6	0.0351	0.0231
7	0.0340	0.0235
8	0.0355	0.0252

No.	C++	Gentee
9	0.0359	0.0229
10	0.0338	0.0243

Berdasarkan data yang ada pada tabel, akan dilakukan uji t 2 sampel independen untuk mendapatkan hasil perbandingan rata-rata eksekusi kedua program. Sebelum melakukan uji t tersebut. Akan diperiksa data di atas menyebar normal atau tidak. Apabila data tidak menyebar normal, maka uji-t 2-sampel tidak tepat diterapkan.

Hipotesis uji kenormalan data adalah sebagai berikut:

H 0 : Data menyebar normal

H 1 : Data tidak menyebar normal

Hasil uji normalitas data dengan menggunakan statistik uji Lilliefors (Kolmogorov-Smirnov). Dengan taraf signifikansi $\alpha = 0.05$.

Gambar 12. *Output Test Normality*

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
C	.162	10	.200 [*]	.895	10	.190
Gentee	.225	10	.162	.936	10	.512

Keluaran pada gambar di atas menunjukkan uji normalitas data C ++ dan Gentee. Pengujian dengan SPSS berdasarkan pada uji Kolmogorov-Smirnov dan Shapiro-Wilk. Berdasarkan hasil uji di atas dapat disimpulkan bahwa untuk uji normalitas data

C++ dan Gentee adalah TERIMA H_0 , karena $p\text{-value} > 0.05$. Dengan kata lain, kedua data menyebar normal.

Setelah mengetahui bahwa kedua data tersebut berdistribusi normal, selanjutnya akan dilakukan uji t dengan software SPSS.

Gambar 13. Output Independent Sample Test

Independent Samples Test									
		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	
Data	Equal variances assumed	.440	.516	14.714	18	.000	.010420	.000708	.008932 .011908
	Equal variances not assumed			14.714	17.335	.000	.010420	.000708	.008928 .011912

Group Statistics				
Program	N	Mean	Std. Deviation	Std. Error Mean
Data C	10	.03401	.001732	.000548
Gentee	10	.02359	.001420	.000449

Untuk uji homogenitas (perbedaan varians). Disana tampak bahwa $F=0.440$ ($p=0.516$) karena p diatas 0.05 , maka dapat dikatakan bahwa tidak ada perbedaan varians pada data hasil eksekusi Gentee dan C++. Karena data homogen maka baca lajur kiri (*equal variance assumed*), terlihat bahwa t hitung -14.714 ($\text{sign}<0.05$), artinya tidak ada perbedaan signifikansi hasil eksekusi Gentee dan C++, hal ini dikarenakan selisihnya Cuma tipis. Tetapi dengan melihat mean pada group statistics dapat dikatakan hasil eksekusi gentee lebih baik dibandingkan kasil eksekusi C++

BAB V

PENUTUP

A. Kesimpulan

1. Hasil Pembuatan program aplikasi perkalian matriks, didapat suatu fakta dari kedua bahasa pemrograman tersebut yaitu penggunaan memori pada aplikasi perkalian matriks dengan menggunakan bahasa Gentee adalah 3 kb, sedangkan penggunaan memori pada aplikasi perkalian matriks dengan menggunakan bahasa C++ adalah 540 kb. Eksekusi kecepatan program menggunakan aplikasi AppTimer. Kecepatan rata-rata program aplikasi perkalian matriks dengan menggunakan bahasa Gentee adalah 0.02359 milolisecond sedangkan rata-rata program aplikasi perkalian matriks dengan menggunakan bahasa C++ adalah 0.03401 milolisecond. Jadi dengan melihat penggunaan memori dan kecepatan program saat dieksekusi gentee masih lebih baik dibandingkan C++.
2. Perbedaan yang mendasar pada Gentee dan C++ yaitu pada Gentee tidak diperlukan tanda titik koma (;) untuk mengakhiri sebuah statement dan tidak menggunakan tanda kurung () untuk penggunaan sebuah fungsi, sedangkan pada C++ digunakan tanda titik koma(;) untuk mengakhiri sebuah statement dan menggunakan tanda kurung () untuk mendefinisikan sebuah fungsi.

B. Saran

Karena masih baru, dokumentasi bahasa gentee masih belum umum, sehingga jarang ada yang mengetahui bahasa pemrograman ini. Oleh karena itu, dokumentasi tentang bahasa gentee agar lebih dipublikasikan dengan cara menterjemahkannya kedalam bahasa Indonesia.



DAFTAR PUSTAKA

<http://faculty.petra.ac.id/thiang/download/dkp/Algoritma%20dan%20Flowchart.doc>

<http://fadlibae.wordpress.com/2010/08/24/latar-belakang-pengembangan-teknologi-informasi-dan-komunikasi-untuk-pendidikan>

<http://id.wikipedia.org/wiki/Faktorial>

<http://ilmukamu.wordpress.com/2011/11/28/tipe-data-pada-c/>

<http://labcsore.blogspot.com/2010/03/struktur-program-c.html>

<http://satriaskyterror.wordpress.com/2010/10/08/perbandingan-bahasa-pemrograman/>

<http://ve511z.wordpress.com/2009/10/10/variabel-dalam-c/>

<http://www.gentee.com>

Akdon dan Ridwan. 2009. *Rumus Dan Data Dalam Analisis Statistik*. Bandung : ALFABETA

Arhami, Muhammad dan Desiani, Anita. 2005. *Pemrograman Matlab*. Yogyakarta : Andi

Dayanti, Erlina dan Indriani, Gina dan Purwanto, Heri. 2005. *Aljabar Linear*. Cirebon : PT. Ercontara Rajawali

Dinar, Muhammad. 2008. *Pengantar Dasar Matematika*. Makassar : FMIPA UNM

Kristanto, Andri. 2009. *Algoritma & Pemrograman dengan C++*. Yogyakarta: Graha Ilmu

Munir, Rinaldi. 2005. *Algoritma Dan Pemrograman Dalam Bahasa Pascal dan C*. Bandung: Informatika Bandung

Raharjo, Budi. 2009. *Pemrograman C++*. Bandung: Informatika

LAMPIRAN

1. Program Aplikasi Perkalian Matriks Dengan Menggunakan Bahasa C++

```
#include<iostream>

#include<iomanip>

using namespace std;

int main() {

int i, j, k, bar_a, kol_a, bar_b, kol_b, mat_a[100][100],mat_b[100][100],

    mat_c[100][100];

    cout<<"Masukkan Jumlah Dan Baris Matriks A"<<endl;

    cout<<"===== "<<endl;

    cout<<"Jumlah baris = ";

    cin>>bar_a;

    cout<<"jumlah kolom = ";

    cin>>kol_a;

    cout<<endl;

    cout<<endl;

    cout<<"Masukkan Jumlah Dan Baris Matriks B"<<endl;

    cout<<"===== "<<endl;

    cout<<"Jumlah baris = ";

    cin>>bar_b;
```

```

cout<<"jumlah kolom = ";

cin>>kol_b;

cout<<endl;

if(kol_a==bar_b) {

    cout<<"Matriks A "<<bar_a<<" X "<<kol_a<<endl;

    cout<<"===== "<<endl;

    for(i=0;i<bar_a;i++){

        for(j=0;j<kol_a;j++){

            cout<<"Masukkan nilai "<<i+1<<". "<<j+1<<"= ";

            cin>>mat_a[i][j];

        }

    }

    cout<<endl;

    cout<<"Jadi Matriks A "<<bar_a<<" X "<<kol_a<<endl;

    cout<<"===== "<<endl;

    cout<<endl;

    for(i=0;i<bar_a;i++){

        for(j=0;j<kol_a;j++){

            cout<<setw(4)<<mat_a[i][j];

        }

        cout<<endl;

    }

}

```

```

        cout<<endl;

        cout<<endl;

        cout<<"Matriks B "<<bar_b<<" X "<<kol_b<<endl;

        cout<<"====="<<endl;

        for(i=0;i<bar_b;i++){

            for(j=0;j<kol_b;j++){

                cout<<"Masukkan nilai "<<i+1<<". "<<j+1<< "= ";

                cin>>mat_b[i][j];

            }

        }

        cout<<endl;

        cout<<endl;

        cout<<"Jadi Matriks B "<<bar_b<<" X "<<kol_b<<endl;

        cout<<"====="<<endl;

        cout<<endl;

        for(i=0;i<bar_b;i++){

            for(j=0;j<kol_b;j++){

                cout<<setw(4)<<mat_b[i][j];

            }

            cout<<endl;

        }

        //perkalian matriks

```

```

for(i=0;i<bar_a;i++){
    for(j=0;j<kol_b;j++){
        mat_c[i][j]=0;
        for(k=0;k<bar_b;k++){
            mat_c[i][j]=mat_c[i][j] + mat_a[i][k]*mat_b[k][j];
        }
    }
}
cout<<endl;
cout<<endl;
cout<<"Jadi Matriks A X B"<<endl;
    cout<<"===== "<<endl;
cout<<endl;
for(i=0;i<bar_a;i++){
    for(j=0;j<kol_b;j++){
        cout<<setw(4)<<mat_c[i][j];
    }
    cout<<endl;
}
}
else {

```



```

        cout<<"Jumlah Baris di kolom Matriks A Tidak Sama Dengan Jumlah
        Baris Matriks B";

    }

    return 0;

}

```

2. Program Aplikasi Perkalian Matriks Dengan Menggunakan Bahasa Gentee

```

func main<main>
{
    str input
    int i j k bar_a kol_a bar_b kol_b
    arr mat_a[100,100]
    arr mat_b[100,100]
    arr mat_c[100,100]

    print("\nMasukkan Jumlah Dan Baris Matriks A\n")

    print("=====UNIVERSITAS ISLAM NEGERI\n")

    print("Jumlah baris = ")
    bar_a = int(conread(input))

    print("Jumlah kolom = ")

    kol_a = int(conread(input))

    print("\n")

    print("\n")
}

```

```

print("Masukkan Jumlah Dan Baris Matriks B\n")

print("=====\n")

print("Jumlah baris = ")

bar_b = int(conread(input))

print("Jumlah kolom = ")

kol_b = int(conread(input))

if kol_a==bar_b
{
    print("\nMatriks A \((bar_a) x \((kol_a)\n")
    print("=====\n")
    for i=0,i<bar_a,i++
    {
        for j=0,j<kol_a,j++
        {
            print("Masukkan nilai \((i+1).\((j+1) = ")
            mat_a[i,j]=int(conread(input))
        }
    }
    print("\n")

    print("\nJadi Matriks A \((bar_a) x \((kol_a)\n")
    print("=====\n")

    for i=0,i<bar_a,i++

```

```

{
    for j=0,j<kol_a,j++
    {
        print("\t(mat_a[i,j])")
    }
    print("\n")
}
print("\n")
print("\n")
print("\nMatriks B \ (bar_b) x \ (kol_b)\n")
print("===== \n")
for i=0,i<bar_b,i++
{
    for j=0,j<kol_b,j++
    {
        print("Masukkan nilai \ (i+1).\ (j+1) =")
        mat_b[i,j]=int(conread(input))
    }
}
print("\n")
print("\nJadi Matriks B \ (bar_b) x \ (kol_b)\n")
print("===== \n")

```

```

for i=0,i<bar_b,i++
{
    for j=0,j<kol_b,j++
    {
        print("\t\t(mat_b[i,j])")
    }
    print("\n")
}
for i=0,i<bar_a,i++
{
    for j=0,j<kol_b,j++
    {
        mat_c[i,j]=0;
        for k=0,k<bar_b,k++
        {
            mat_c[i,j]=mat_c[i,j] + mat_a[i,k]*mat_b[k,j];
        }
    }
}

print("\n")

print("\nJadi Matriks A X B\n")

print("=====\n")

```

```

for i=0,i<bar_a,i++
{
    for j=0,j<kol_b,j++
    {
        print("\t\t(mat_c[i,j])")
    }
    print("\n")
}

else
{
    print("\nJumlah Baris di kolom Matriks A Tidak Sama Dengan Jumlah Baris
Matriks B")

}

getch()
}

```



UNIVERSITAS ISLAM NEGERI
ALAUDDIN
 M A K A S S A R